Rogue Hat Model Documentation

By: Grant Thornton, Micah Richardson, Sai Abhiram Addanki

Statement of Purpose

The purpose of this model is to predict an individual loan status, specifically if they would pay their loan back or not based on collected application features such as month, annual income, debt-to-income ratio, and loan amount.

From a business use standpoint, this model will help aid financial institutions like Lending Tree with identifying high-risk applicants or guiding decision-making for typical profiles that should be approved for a credit loan. By integrating the model into the loan evaluation process improvements in decision-making speed, reductions in default risks, and overall benefits in responsible lending practices will be ensured.

Foundational Data

Initial preprocessing was performed by following the "Starter_Notebook" provided. The data was retrieved from https://github.com/fiddler-labs/p2p-lending-data/tree/refs/heads/master/raw_data and initially had 2,132,287 observations. The target variable "loan_status" was identified and then subset to only rows where the loan was either fully paid off or not. Only loans from August 1st 2012 were kept for analysis. Next dummy encoding was applied to categorical features before applying null handling on missing values for all columns except "fully_paid", "issue_d", "zip_code", "addr_state". From there the data was subset to only contain 140910 rows.

Features/Data Dictionary

In total, there were only 40 predictors that were included in the final model.

acc_open_past_24mths: Number of trades opened in past 24 months.

annual_inc: The self-reported annual income provided by the borrower during registration.

avg_cur_bal: Average current balance of all accounts.

bc_open_to_buy: Total open to buy on revolving bankcards.

bc_util: Ratio of total current balance to high credit/credit limit for all bankcard accounts.

dti: A ratio calculated using the borrower's total monthly debt payments on total debt obligations, excluding mortgage and requested LC loan, divided by the borrower's self-reported monthly income.

fico_range_high: The upper boundary range the borrower's FICO at loan origination belongs to.

fico_range_low: The lower boundary range the borrower's FICO at loan origination belongs to.

mo_sin_old_rev_tl_op: Months since oldest revolving account opened.

mo_sin_rcnt_tl: Months since most recent account opened.

mort_acc: Number of mortgage accounts.

mths_since_last_delinq: The number of months since the borrower's last delinquency.

mths_since_last_major_derog: Months since most recent 90-day or worse rating.

mths_since_last_record: The number of months since the last public record.

mths_since_recent_bc: Months since most recent bankcard account opened.

mths_since_recent_bc_dlq: Months since most recent bankcard delinquency.

mths_since_recent_inq: Months since most recent inquiry.

num_actv_bc_tl: Number of currently active bankcard accounts.

num_actv_rev_tl: Number of currently active revolving trades.

num_il_tl: Number of installment accounts.

num_rev_tl_bal_gt_0: Number of revolving trades with balance > 0.

num_tl_90g_dpd_24m: Number of accounts 90 or more days past due in the last 24 months.

num_tl_op_past_12m: Number of accounts opened in past 12 months.

percent_bc_gt_75: Percentage of all bankcard accounts > 75% of limit.

revol_bal: Total credit revolving balance.

revol_util: Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.

sec_app_fico_range_high: The upper boundary range the co-borrower's FICO score belongs to.

sec_app_fico_range_low: The lower boundary range the co-borrower's FICO score belongs to.

tot_cur_bal: Total current balance of all accounts.

tot_hi_cred_lim: Total high credit/credit limit.

total_acc: The total number of credit lines currently in the borrower's credit file.

total_bc_limit: Total bankcard high credit/credit limit.

total_il_high_credit_limit: Total installment high credit/credit limit.

total_rev_hi_lim: Total revolving high credit/credit limit.

credit_line_age: The calculated age of the borrower's credit line.

verification_status_Verified: Indicates if income was verified by LC.

month: The month in which the loan was funded.

Feature Engineering

For time-based columns like mo_sin and acc_open, feature engineering was implemented to create value groupings and mappings of new classes. Missing values were filled based on feature distributions as implemented in the notebook.

Additionally, the target feature for loan status was dropped and replaced with the new feature "fully_paid" where 'Fully Paid' status was 1 and 'Charged Off' status was 0.

Splitting and Training

For all of the continuous features, they were split into the x train variable and any remaining missing values were imputed with its mean value. For any remaining null values, a linear regression model was then trained using the non-missing rows of the target feature and the predictor variables. This trained model was then used to predict the missing values in the target feature, and those predictions replaced the null values. The response variable and non-continuous features were excluded from this process to avoid leakage or bias.

Feature Outliers

To account for variations in feature distribution, z-scores were calculated for each feature and any values above our threshold were removed from the dataset.

Hyperparameter Optimization

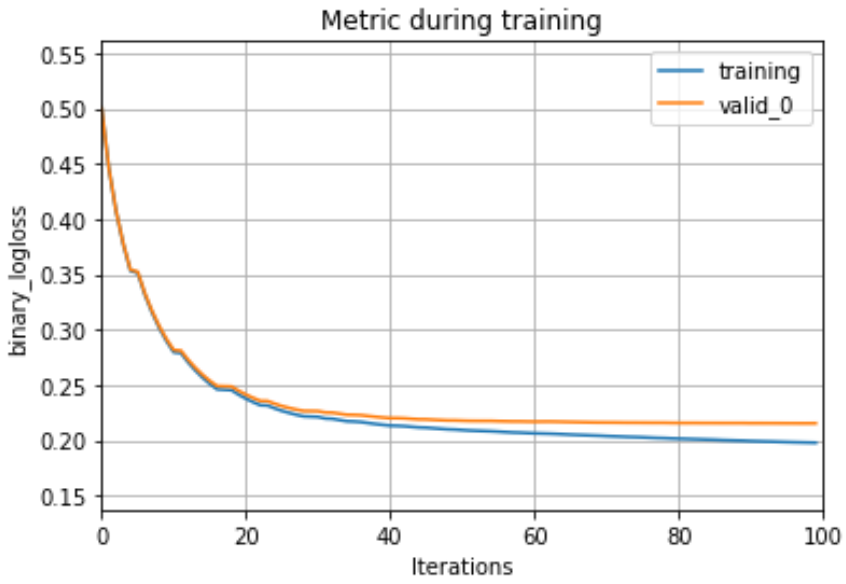
Hyperparameter optimization for the models was achieved using the Optuna library. The hyperparameters are defined within specific ranges using the trial.suggest_ methods provided by Optuna. The model is trained on a subset of the data, and its performance is evaluated on a validation set. The evaluation metric used is accuracy, calculated by comparing the model's predictions to the true labels. The primary goal of this process is to optimize the accuracy of the binary classification task and output the best hyperparameter values for the Light Gradient Boosting Machine(LGBM) and eXtreme Gradient Boosting(XGB) model.

Model Training and Evaluation

A LightGBM regressor and XGB was trained on the entire training set using the best hyperparameters obtained from the optimization process. The model was then evaluated on the testing set by calculating and comparing the training and test accuracy scores.

Model Comparison

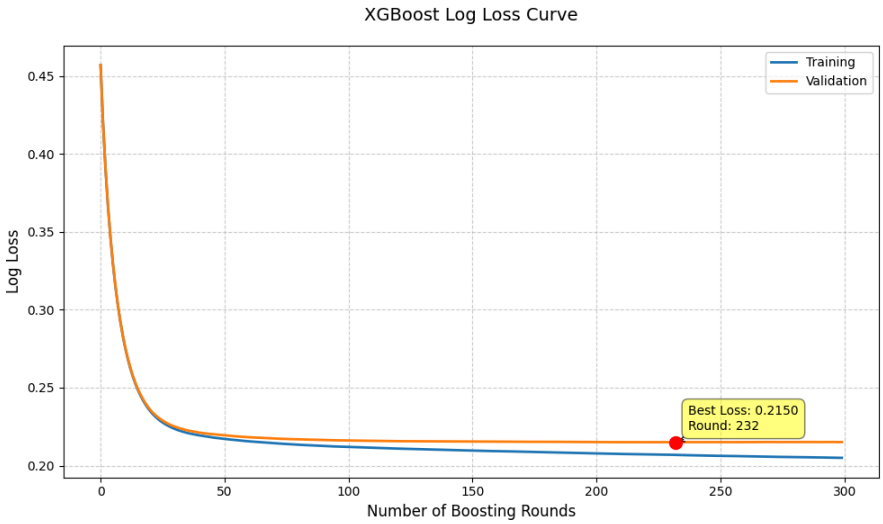
LGBM Model 1:



Training accuracy 0.9170

Testing accuracy 0.9088

XGBoost Model 2:

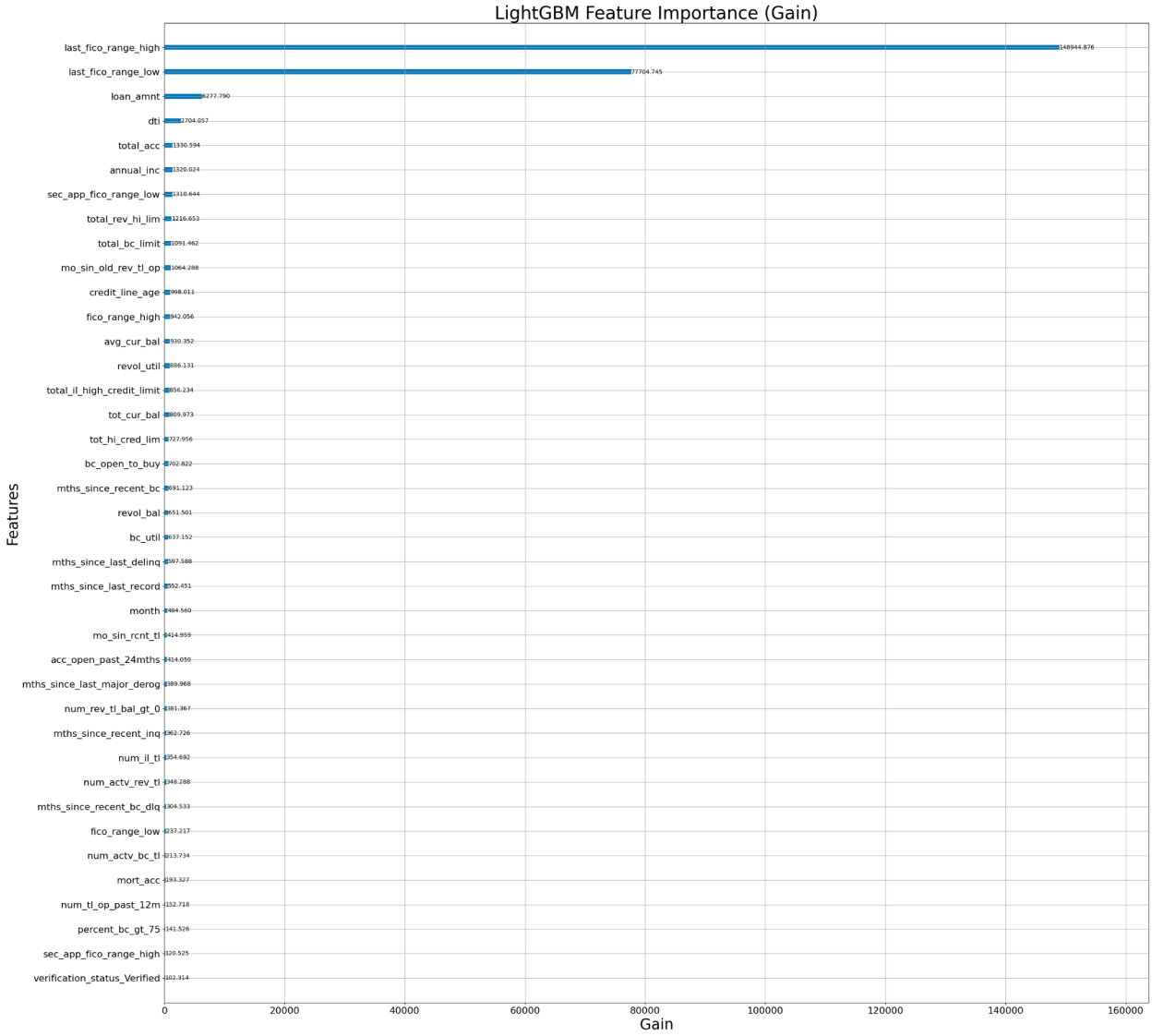


Training accuracy 0.9139

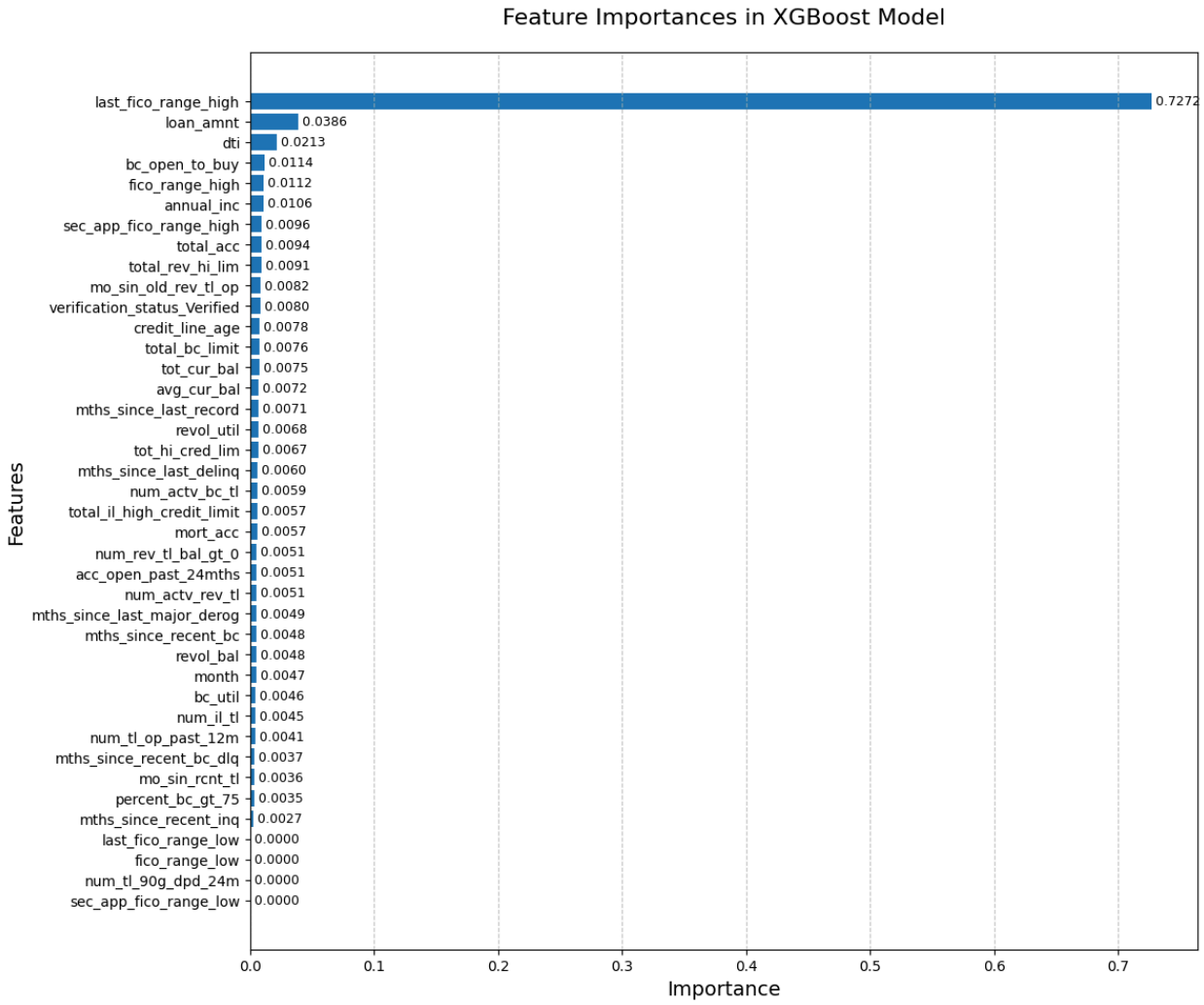
Testing accuracy 0.9092

XGBoost Model 2 is slightly better than LGBM Model 1 because it has marginally better testing accuracy and shows slightly better generalization, meaning it performs better on unseen data. Given its ability to generalize better on unseen data, we suggest that we move forward with the XGBoost model.

Feature Importance LGBM Model 1:

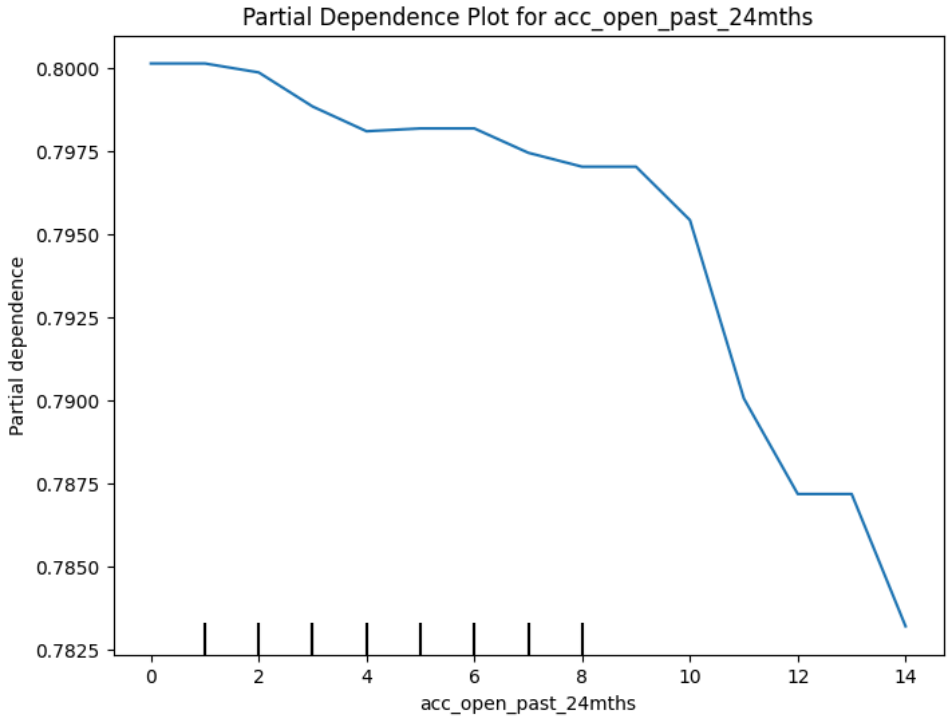


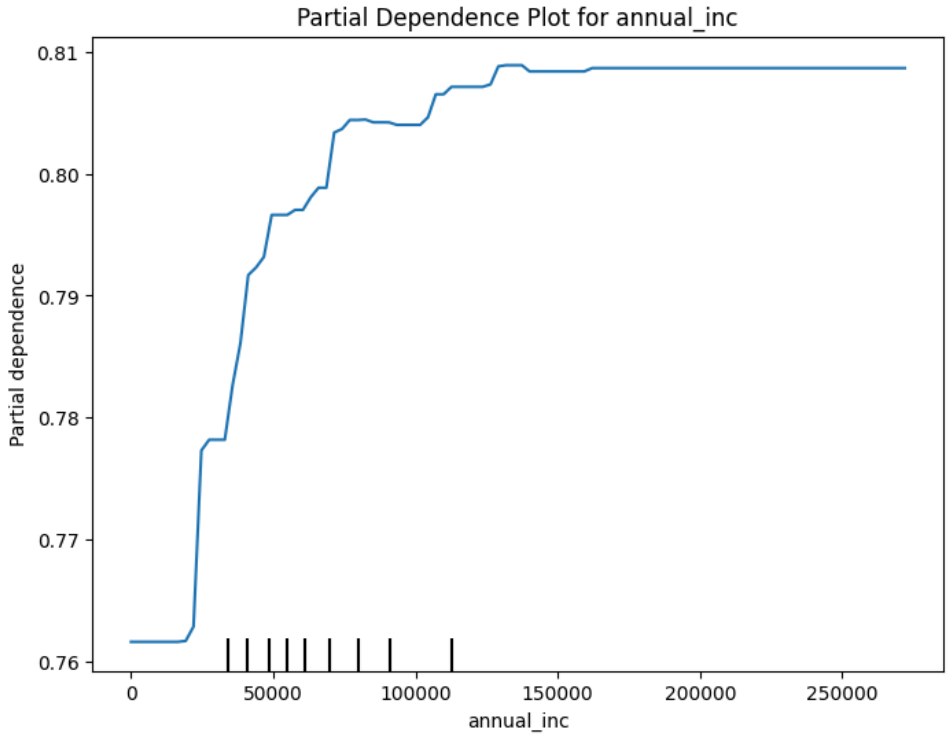
Feature Importance XGBoost Model 2:

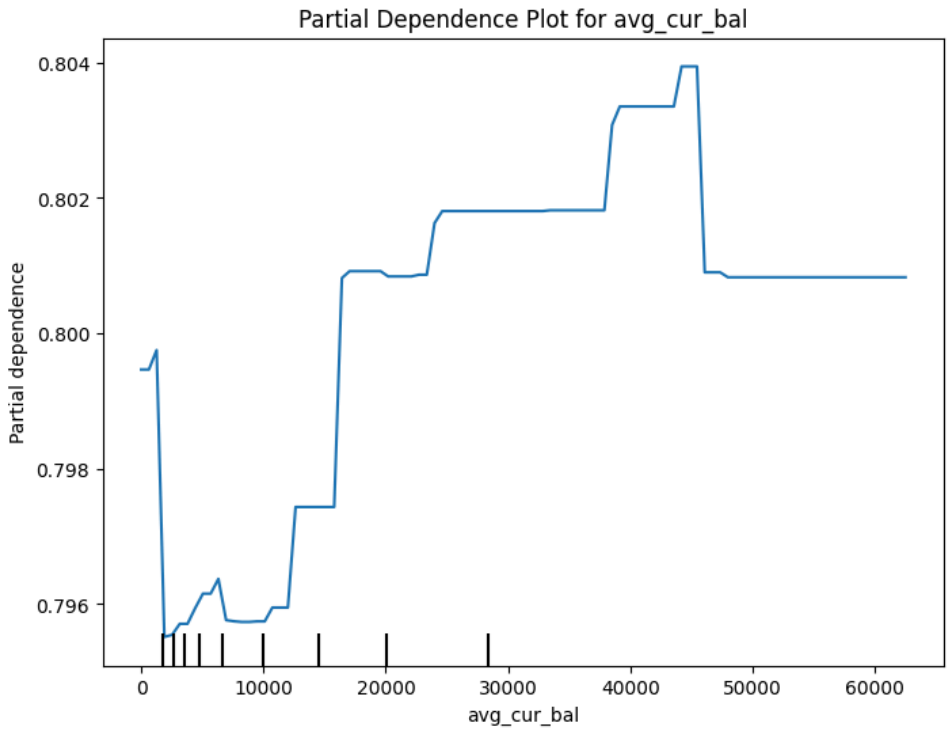


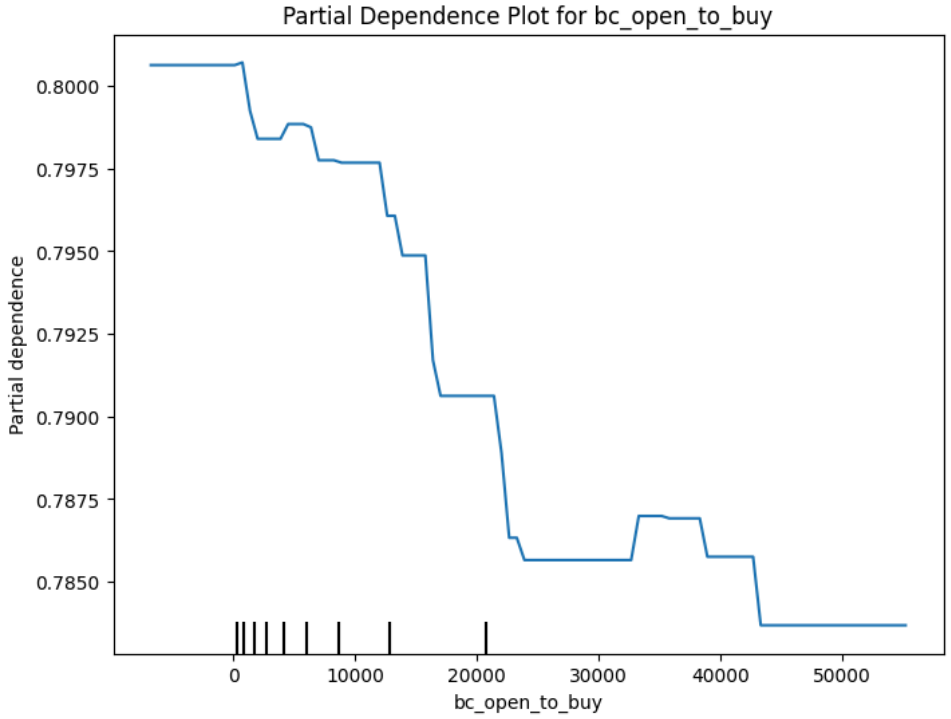
Partial Dependence plots

The notebook includes the output to the partial dependence plots of each predictor.






Partial Dependence Plot for avg_cur_bal
Partial dependence
0.804
0.802
0.800
0.798
0.796
0
10000
20000
30000
40000
50000
60000
avg_cur_bal


Partial Dependence Plot for bc_open_to_buy
Partial dependence
0.8000
0.7975
0.7950
0.7925
0.7900
0.7875
0.7850
0
10000
20000
30000
40000
50000
bc_open_to_buy

Weaknesses/risks with model

While the XGBoost model performed better than the LGBM model, there are still several risks that inherently come with its use. First the model's complexity can have issues with overfitting, especially given the structure of the dataset used in this business scenario. In general if there are issues with hyperparameter tuning, additional overfitting issues will occur. As for weaknesses with the XGBoost model, it requires a significant amount of computational power in terms of memory and processing. Additionally, feature importance may not solely provide enough interpretability given the use case and can be a drawback in a highly regulated modeling field like finance.

As for weaknesses and risks for the data preparation of the model, there are a few areas of concern. First outliers were identified using z-scores, but this method assumes normality in feature distributions. If features have non-normal distributions, important extreme data points might have been removed, reducing the model's insight. Example could be a feature with a large proportion of high net worth applicants or loan amounts, and the information could have been reduced as an impact

Additionally with the data cleaning process, data prior to August 2012 was omitted from the model. This was largely due to missing values however there is a risk that the model output is overfitted to the training data from this chosen time period and misses out on trends prior to August 2012. As a subsequent result, this could lead to a weakness with the results of the Optuna hyperparameter tuning and model outputs.

Controls

A potential control for this model would be to continuously monitor the distributions of key features over time, which could indicate a mismatch between training data and real-world trends. Additionally, periodic retraining (ex. every quarter) could ensure that market conditions are being captured properly and if there any changes in application feature impacts.

Another control could be to incorporate additional evaluation metrics to ensure the model balances the right objectives according to the business focus. For example, precision may be a better metric to minimize risk of charge offs. Recall may be better if the business is trying to focus on gaining more profit by maximizing loan approvals for the right applicants. F1-score could also be a useful metric in replacement of accuracy as a general evaluation method.